Claims

1. A method of implementing a software application on a target hardware architecture having a first and second processing resource and a predetermined interaction protocol between the first and second processing resources, comprising:

designing a software system independent of the target hardware architecture, the software system including a first and a second functional object each performing a predetermined action, a coordination object for regulating control and dataflow interactions between the first and second functional objects, a control object for handling control interactions between one of the functional objects and the coordination object, and a dataflow object for handling dataflow interactions between one of the functional objects and the coordination object;

creating a software graph based on the software system, in which the first functional object is represented as a first set of software nodes in which a first action and a first mode within the first functional object are represented as a first action node and a first mode node, respectively, and in which the second functional object is represented as a second set of software nodes, in which a second action and a second mode within the second functional object are represented as a second action node and a second mode node, respectively, and in which the coordination object is represented as a set of coordination nodes in which a coordination action is represented as a coordination action node and a coordination mode is represented as a coordination mode node, and in which the dataflow object is represented as a dataflow edge connecting one of the software nodes within the first and second sets of software nodes to one of the coordination nodes within the set of coordination nodes, and the control object is represented as a control edge connecting one of software nodes within the first and second sets of software nodes to one of the coordination nodes within the set of coordination nodes;

creating a hardware graph based on the target hardware architecture wherein the first processing resource is represented as a first hardware node, the second processing resource is represented as a second hardware node and the interaction protocol is represented as a hardware edge connecting the first and second hardware nodes; and

mapping the software graph to the hardware graph, wherein the first set of software nodes is mapped to the first hardware node, the second set of software nodes is mapped to the second hardware node, the set of coordination nodes are mapped to the first hardware node and one of the control and dataflow edges is mapped to the hardware edge.

2. A method according to claim 1 further comprising replacing the one of the control and dataflow edges that has been mapped to the hardware edge with a first replacement node mapped to the first hardware node and a second replacement node mapped to the second hardware node, for implementing said edge in terms of the interaction protocol provided by the target hardware architecture.

3. A method according to claim 2 in which the first and second replacement nodes are chosen from a set of replacement nodes based upon the type of edge being replaced and the interaction protocol provided by the target hardware architecture.

4. A method according to claim 3 further comprising generating implementation code for the first processing resource based on the first set of software nodes , the set of coordination nodes, and the first replacement node that were mapped to the first hardware node, and generating implementation code for the second processing resource based on the second set of software nodes that were mapped to the second hardware node.

5. A method according to claim 1 of implementing the software system on a second target hardware architecture having three processing resources and a predetermined first interaction protocol between a first and a second processing resource of the three processing resources and a predetermined second interaction protocols between the second and a third of the three processing resources further comprising:

creating a second hardware graph based on the second target hardware architecture wherein the first of the three processing resource is represented as a first hardware node, the second processing resource of the three processing resources is represented as a second hardware node, the third processing resource of the three processing resources is represented as a third hardware node, the predetermined first interaction protocol is represented as a first hardware edge connecting the first and second hardware nodes, and the predetermined second interaction protocol is represented as a second hardware edge connecting the second and third hardware nodes;

mapping the software graph to the second hardware graph, wherein the first set of software nodes are mapped to the first hardware node, the second set of software nodes are mapped to the third hardware node, the set of coordination nodes are mapped to the second hardware node, a crossing dataflow edge connecting one of the coordination nodes to one of the software nodes of the first set of software nodes is mapped to the first hardware edge, and a crossing control edge connecting one of the coordination nodes to one of the software nodes of the second set of software nodes is mapped to the second hardware edge;

replacing the crossing dataflow edge with a first data replacement node mapped to the first hardware node and a second data replacement node mapped to the second hardware node for implementing the crossing dataflow edge in terms of the predetermined first interaction protocol; and

replacing the crossing control edge with a first control replacement node mapped to the second hardware node, and a second control replacement node mapped to the third hardware node for implementing the crossing control edge in terms of the predetermined second interaction protocol.

6. A method according to claim 5 in which in which the first and second data replacement nodes and the first and second control replacement nodes are chosen from a set of replacement nodes based upon the interaction protocols provided by the target hardware architecture.

7. A method according to claim 6 further comprising:

generating implementation code for the first processing resource based on the first set of software nodes and the first control replacement node;

generating implementation code for the second processing resource based on the set of coordination nodes, the second data replacement node, and the first control replacement node; and

generating implementation code for the third processing resource based on the second set of software nodes and the second control replacement node.

8. A system for generating an executable version of a software application for use with a target hardware architecture having a plurality of processing resources and an interaction protocol for controlling information exchanges between the processing resources, the software application being designed independent of the target hardware architecture, the system comprising:

a software application design tool for building software applications based on functional requirements and explicit coordination characteristics;

a software graphing tool that generates a coordination graph based on the software application, in which first and second functional components of the software application are represented as first and second software nodes, respectively, first and second software nodes having a node type that is based on a function performed by the respective functional component, and in which an information exchange between the first and second functional components is represented as a software edge connecting the first and second software nodes, and the software edge has an edge type that is based on the type of information exchange it represents;

a hardware graphing tool for generating a target hardware graph based on the target hardware architecture, in which first and second computational resources are represented as first and second hardware nodes, respectively, and the interaction protocol which controls information exchanges between the processing resources is represented as a hardware edge connecting the first and second software nodes;

a mapping tool in which the first and second software nodes are mapped to first and second hardware nodes, respectively, and the software edge is mapped to the hardware edge, and first and second replacement nodes are mapped to first and second hardware nodes, each replacement node representing a functional replacement component that will implement the information exchange represented by the software edge in terms of the interaction protocol represented by the hardware edge, and will pass any interactions to the respective first and second components; and

a code synthesizing tool in which an executable version of the first functional and functional replacement components will be generated for use on the first processing resource and an executable version of the second functional and functional replacement components will be generated for use on the second processing resource.